WE CLAIM:

1. A multithreaded processor, comprising:

a fetch control unit, having an input coupled to receive an execution feedback signal with information related to a plurality of threads on a per thread basis, the fetch control unit generating an instruction fetch sequence based on the execution feedback signal; and

an instruction cache, having an input coupled to an output of the fetch control unit, the instruction cache dispatching instruction data responsive to the instruction fetch sequence.

2. The multithreaded processor of claim 1, further comprising an instruction queue having an output coupled to the fetch control unit input, the instruction queue generating a queue feedback signal responsive to a thread queue condition associated with a thread from the plurality of threads, wherein the fetch control unit generates the instruction fetch sequence also based on the queue feedback signal.

3. The multithreaded processor of claim 2, wherein the thread queue condition indicates that a thread queue has less than a first amount of remaining storage.

4. The multithreaded processor of claim 2, wherein the thread queue condition indicates that a thread queue has less than a second amount of remaining decoded instructions.

5. The multithreaded processor of claim 1, wherein the fetch control unit blocks the thread from the instruction fetch sequence responsive to the queue feedback signal.

6. The multithreaded processor of claim 1, wherein the fetch control unit advances the thread in the instruction fetch sequence responsive to the queue feedback signal.

7. The multithreaded processor of claim 1, further comprising a thread interleaver having an output coupled to the fetch control unit input, the thread interleaver

generating an interleaver feedback signal responsive to a thread condition, wherein the fetch control unit generates the instruction fetch sequence also based on the interleaver feedback signal.

8. The multithreaded processor of claim 1, wherein the thread condition indicates that a thread from the plurality of threads is ineligible for execution.

9. The multithreaded processor of claim 1, wherein the thread interleaver generates a thread execution sequence independent of the instruction fetch sequence.

10. The multithreaded processor of claim 1, further comprising an execution pipeline having an output coupled to the fetch control unit input, the execution pipeline generating the execution feedback signal responsive to an execution stall.

11. The multithreaded processor of claim 1, wherein the fetch control unit delays the thread in the instruction fetch sequence responsive to the execution stall.

12. The multithreaded processor of claim 1, wherein the execution stall comprises one from the group consisting of a branch misprediction, an exception, a data cache miss, an external resource stall, an interlock, and a memory operation ordering.

13. The multithreaded processor of claim 1, wherein the fetch control unit generates the instruction fetch sequence, in a default state, by selecting a thread from the plurality of threads according to round robin arbitration.

14. The multithreaded processor of claim 1, wherein the execution feedback signal is capable of including information related to each of the plurality of threads.

15. The multithreaded processor of claim 1, wherein the multithreaded processor is a multithreaded network processor and the instruction data are packet processing instructions related to at least one form the group consisting of: packet routing, switching, bridging and forwarding.

16. A multithreaded processor, comprising:

means for fetching, having an input coupled to receive an execution feedback signal with information related to a plurality of threads on a per thread basis, the means for fetching generating an instruction fetch sequence based on the execution feedback signal; and

means for storing, having an input coupled to an output of the means for fetching, the means for storing dispatching instruction data responsive to the instruction fetch sequence.

17. The multithreaded processor of claim 16, further comprising a means for queuing having an output coupled to the means for fetching input, the means for queuing generating a queue feedback signal responsive to a thread queue condition associated with a thread from the plurality of threads, wherein the means for fetching generates the instruction fetch sequence also based on the queue feedback signal.

18. The multithreaded processor of claim 17, wherein the thread queue condition indicates that a means for thread queuing has less than a first amount of remaining storage.

19. The multithreaded processor of claim 17, wherein the thread queue condition indicates that a means for thread queuing has less than a second amount of remaining decoded instructions.

20. The multithreaded processor of claim 16, wherein the means for fetching blocks the thread from the instruction fetch sequence responsive to the queue feedback signal.

21. The multithreaded processor of claim 20, wherein the means for fetching advances the thread in the instruction fetch sequence responsive to the queue feedback signal.

22. The multithreaded processor of claim 16, further comprising a means for interleaving having an output coupled to the means for fetching input, the means for interleaving generating an interleaver feedback signal responsive to a thread condition, wherein the means for fetching generates the instruction fetch sequence also based on the interleaver feedback signal.

23. The multithreaded processor of claim 16, wherein the thread condition indicates that a thread from the plurality of threads is ineligible for execution.

24. The multithreaded processor of claim 16, wherein the means for interleaving generates a thread execution sequence independent of the instruction fetch sequence.

25. The multithreaded processor of claim 16, further comprising an means for executing having an output coupled to the means for fetching input, the means for executing generating the execution feedback signal responsive to an execution stall.

26. The multithreaded processor of claim 16, wherein the means for fetching delays the thread in the instruction fetch sequence responsive to the execution stall.

27. The multithreaded processor of claim 16, wherein the execution stall comprises one from the group consisting of a branch misprediction, an exception, a data cache miss, an external resource stall, an interlock, and a memory operation ordering.

28. The multithreaded processor of claim 16, wherein the means for fetching generates the instruction fetch sequence, in a default state, by selecting a thread from the plurality of threads according to round robin arbitration.

29. The multithreaded processor of claim 16, wherein the execution feedback signal is capable of including information related to each of the plurality of threads.

30. The multithreaded processor of claim 16, wherein the multithreaded processor is a multithreaded network processor and the instruction data are packet processing

instructions related to at least one form the group consisting of: packet routing, switching, bridging and forwarding.

31. A method for fetching instructions in a multithreaded processor, comprising:

generating an instruction fetch sequence based on an execution feedback signal with information related to a plurality of threads on a per thread basis; and

dispatching instruction data responsive to the instruction fetch sequence.

32. The method of claim 31, further comprising

generating a queue feedback signal responsive to a thread queue condition associated with a thread from the plurality of threads.

33. The method of claim 32, wherein the thread queue condition indicates that a thread queue has less than a first amount of remaining storage.

34. The method of claim 32, wherein the thread queue condition indicates that a thread queue has less than a second amount of remaining decoded instructions.

35. The method of claim 31, wherein the generating the instruction fetch sequence comprises blocking the thread from the instruction fetch sequence responsive to the queue feedback signal.

36. The method of claim 35, wherein the generating the instruction fetch sequence comprises advancing the thread in the instruction fetch sequence responsive to the queue feedback signal.

37. The method of claim 31, wherein the generating the instruction fetch sequence further comprises generating an interleaver feedback signal responsive to a thread condition in a thread interleaver.

38. The method of claim 31, wherein the thread condition indicates that a thread from the plurality of threads is ineligible for execution.

39. The method of claim 31, wherein the generating the feedback signal comprises generating a thread execution sequence independent of the instruction fetch sequence.

40. The method of claim 31, the generating the instruction fetch sequence further comprises generating the execution feedback signal responsive to an execution stall.

41. The method of claim 31, wherein the generating the instruction fetch sequence further comprises delaying the thread in the instruction fetch sequence responsive to the execution stall.

42. The method of claim 31, wherein the execution stall comprises one from the group consisting of a data cache miss, an external resource stall, an interlock, and a memory operation ordering.

43. The method of claim 31, wherein the generating the instruction fetch sequence comprises generating the instruction fetch sequence, in a default state, by selecting a thread from the plurality of threads according to round robin arbitration.

44. The method of claim 31, wherein the feedback signal is capable of including information related to each of the plurality of threads.

45. The method of claim 31, wherein the multithreaded processor is a multithreaded network processor and the instruction data are packet processing instructions related to at least one form the group consisting of: packet routing, switching, bridging and forwarding.